REMARKS

Claims 1-21 are pending in this application. Claims 1, 4, 11, 12, 18 and are amended. Claims 5 and 19 are canceled without prejudice or disclaimer. Claim 21 is new.

No new matter has been added. The amendments are supported by Applicant's Specification, including original claim 4, page 4, line 6; page 6, line 29; page 9, line 5; page 19, lines 12-15; page 18, lines 13-15 and 18-20; and page 23, lines 4-5 and 20.

The amendments clarify that the input stream has a tag structure with start and end tags, which may enclose further tags, so that the tag structure may be represented by a tree structure. Also, the amendments clarify that the objects mapped to tags define start and end methods, which can be invoked. Further, the amendments clarify that, in response to finding a start tag, the start method defined in the object mapped to the tag found is invoked and, in response to finding an end tag, the end method defined in the object mapped to the tag found is invoked. In addition, the amendments clarify that, due to the fact that the object is built into the tree representation before one of the methods defined by it is invoked, the method invoked has awareness of the position of the tag to which it is mapped within the tree structure, as the tree structure is built.

The Office Action rejected claims 1-5, 7-11, 12-17, and 18-20 under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,125,391 to Meltzer et al. ("Meltzer").

Applicant respectfully traverses the rejection, because Meltzer does not teach each and every claim element as arranged in the claims as interpreted by one of ordinary skill in the art.

Meltzer discloses a method of processing extended markup language (XML) documents according to which the document is parsed and, when a parsing event (e.g., a

certain tag) occurs, the parser calls a particular method in an application object. (See, e.g., Meltzer, col. 26, lines 59-61). The method called is not aware of the parsing event's (or tag's) position in a tree structure representation of the XML document, nor even of its position in the stream of XML tags in the input document. It is clear for the skilled person that there is no such awareness, not only because it is not mentioned in Meltzer, but, moreover, by Meltzer disclosing that the parser can be implemented by using a "standard call back model." (Meltzer, col. 26, lines 59-61). The skilled person knows that an XML event call back model lists only the tag name and its attributes and, therefore, has no tree-structure awareness, nor even position-in-stream awareness.

Although in Meltzer a tree representation of the input document is also generated (col. 28, lines 8-23), the application objects bound to the tags are not built into the tree to provide the methods with tree-position awareness, as the tree is built. This can, for example, be seen from the fact that Fig. 5 does not, in any way, show that 503, 504, or 505 are aware of the structure being built at 506 as it is being built. Rather, in Meltzer, the tree structure must first be built before it can be used, for example, to generate XML events, in a second pass, which could then be used as input XML events. (See, for example, col. 28, lines 16-21 and Fig. 5, which implies this by the loop form 507-511). Thus, as the input XML document is parsed and the tree is built, the methods called have no tree-position awareness.

Comparing now the teaching of Meltzer with the amended claim 1, it is clear that Meltzer does not disclose at least the following elements.

Claim 1 recites, *inter alia*, "as a tag is found during the parsing process, building a tree representation of the input stream and the objects bound to tags by building the object mapped to the tag found into the tree representation according to the tag structure". According to claim 1, as a tag is found during the parsing process, a tree representation of the input stream and the objects bound to tags is built. In Meltzer, however, only a tree representation of the input stream is built, with no objects bound to tags. According to

claim 1, the tree representation is built by building the object mapped to the tag found into the tree representation according to the tag structure. In Meltzer, however, no objects mapped to the tag found are built into the tree representation.

Claim 1 recites, *inter alia*, "wherein, due to the fact that the object is built into the tree representation before one of the methods defined by it is invoked, the method invoked has awareness of the position of the tag to which it is mapped within the tree structure, as the tree structure is built." According to claim 1, as a tag is parsed, the object is built into the tree representation before the method mapped to the currently parsed tag is invoked. In Meltzer, however, it is not disclosed that the tree representation is built before the method mapped to the currently parsed tag is invoked. According to claim 1, the method invoked has awareness of the position of the tag to which it is mapped within the tree structure, as the tree structure is built. In Meltzer, however, the method invoked has no awareness of the tree structure position, as the tree structure is built.

The awareness of the tree-structure position of the method invoked according to the claimed invention has the advantage that it enables the method not only to depend on the parameters of the respective tag (as in Meltzer), but also to depend on the tag's position within the tree structure (i.e., on parent, child, and sibling relationships with other tags), which increases the power of a markup language used to control "behavior." According to the claimed invention, this awareness is already obtained during the first pass, as the tags are found, before building the entire tree structure has been completed. This is not achieved by the method disclosed by Meltzer, as explained above. In addition, Walker does not disclose, whatsoever, any hint to this solution.

For the same reasons given above with respect to claim 1, claims 12, 18, and 21 are also patentable over Meltzer. Also, dependent claims 2-4, 7-11, 13-17, and 20 depend, directly or indirectly, from one of the independent claims 1, 12, 18, and 21 and, thus,

inherit their patentable subject matter. Therefore, claims 1-4, 7-11, 12-17, 18, and 20-21 are patentable over Meltzer. (Claims 5 and 19 are canceled.)

The Office Action rejected claim 6 under 35 U.S.C. §103(a) as being unpatentable over Meltzer in view of U.S. Patent No. 6,434,529 to Walker et al. ("Walker").

For the same reasons given above with respect to claim 1, claim 6, which depends from claim 1 and, thus, inherits the patentable subject matter of claim 1, is patentable over Meltzer and Walker.

In view of the foregoing, Applicant respectfully submits that all of the claims in the present application are patentably distinguishable over the reference cited in the Office Action. Accordingly, Applicant respectfully requests reconsideration and that the claims be passed to allowance.

Respectfully submitted,

2-17-04
Date

Paul D. Greeley, Esq.
Reg. No. 31019
Attorney for the Applicant
Ohlandt, Greeley, Ruggiero & Perle, L.L.P.
One Landmark Square, 10th Floor
Stamford, CT 06901-2682
Tel: 203-327-4500
Fax: 203-327-6401